CLAIMS:

What is claimed is:

1. A method of event handling in a multiprocessor system, comprising:

encountering the event by one processor in the multiprocessor system;

stopping at least one other processor of the multiprocessor system by forcing the at least one other processor to fetch instructions from a copy of an operating system kernel having at least one breakpoint.

2. The method of claim 1, further comprising:

copying an original version of an operating system kernel to a new physical location of memory in order to generate a new version of the operating system kernel;

switching address mapping of the multiprocessor system to direct mapping to the new version of the operation system kernel; and

inserting the at least one breakpoint in the original version of the operating system kernel to generate the copy of the operating system kernel having at least one breakpoint.

3. The method of claim 1, further comprising:

causing one or more caches of the at least one other processor of the multiprocessor system to be flushed to thereby force the at least one other processor to fetch instructions.

4. The method of claim 1, wherein the at least one other processor encounters the at least one breakpoint and enters a breakpoint handler of the copy of the operating system kernel, and wherein the processors are redirected to a debugger.

5. The method of claim 2, wherein the steps of switching address mapping and inserting at least one breakpoint are performed in response to the occurrence of an event.

6. The method of claim 5, wherein the event is encountering one of a breakpoint, a trigger, a watchpoint, or the occurrence of an error.

7. The method of claim 4, wherein the breakpoint handler is unmodified from the breakpoint handler in the original copy of the operating system kernel.

8. The method of claim 2, wherein inserting at least one breakpoint in the original copy of the original operating system kernel includes inserting a repeating pattern of breakpoint instruction opcodes into all portions of the original copy of the operating system kernel with the exception of a breakpoint handler in the original copy of the operating system kernel.

9. The method of claim 1, wherein the multiprocessor system is a multiprocessor system that does not support the use of non-maskable interrupts.

10. The method of claim 1, wherein the multiprocessor system is one of a symmetric multiprocessor system and a non-uniform memory access system.

11. An apparatus for handling an event in a multiprocessor system, comprising:

means for copying an original copy of an operating system kernel to a new physical location of memory in order to generate a new copy of the operating system kernel;

means for switching address mapping of the multiprocessor system to direct mapping to the new copy of the operation system kernel;

means for inserting at least one breakpoint in the original copy of the operating system kernel to generate a modified copy of the operating system kernel, wherein when a processor of the multiprocessor system is forced to fetch an instruction from the modified copy of the operating system kernel, the processor is redirected to the new copy of the operating system kernel to thereby handle the event.

12. The apparatus of claim 11, further comprising:

means for causing one or more caches of processors of the multiprocessor system to be flushed.

13. The apparatus of claim 12, wherein the processors of the multiprocessor system refetch instructions from the modified copy of the operating system kernel and encounter the inserted at least one breakpoint.

14. The apparatus of claim 13, wherein the processors enter a breakpoint handler of the modified copy of the operating system kernel, and wherein the processors are redirected to a debugger.

15. The apparatus of claim 11, wherein the means for switching address mapping and means for inserting at least one breakpoint are operable in response to the occurrence of an event.

16. The apparatus of claim 15, wherein the event is encountering one of a breakpoint, a trigger, a watchpoint, or the occurrence of an error.

17. The apparatus of claim 14, wherein the breakpoint handler is unmodified from the breakpoint handler in the original copy of the operating system kernel.

18. The apparatus of claim 11, wherein the means for inserting at least one breakpoint in the original copy of the original operating system kernel includes means for inserting a repeating pattern of breakpoint instruction opcodes into all portions of the original copy of the operating system kernel with the exception of a breakpoint handler in the original copy of the operating system kernel.

19. The apparatus of claim 11, wherein the multiprocessor system is a multiprocessor system that does not support the use of non-maskable interrupts.

20. The apparatus of claim 11, wherein the multiprocessor system is one of a symmetric multiprocessor system and a non-uniform memory access system.

21. A computer program product in a computer readable medium for handling an event in a multiprocessor system, comprising:

first instructions for copying an original copy of an operating system kernel to a new physical location of memory in order to generate a new copy of the operating system kernel;

second instructions for switching address mapping of the multiprocessor system to direct mapping to the new copy of the operation system kernel; and

third instructions for inserting at least one breakpoint in the original copy of the operating system kernel to generate a modified copy of the operating system kernel, wherein when a processor of the multiprocessor system is forced to fetch an instruction from the modified copy of the operating system kernel, the processor is redirected to the new copy of the operating system kernel to thereby handle the event.

22. The computer program product of claim 21, further comprising:

fourth instructions for causing one or more caches of processors of the multiprocessor system to be flushed.

23. The computer program product of claim 21, wherein the third instructions include instructions for inserting a repeating pattern of breakpoint instruction opcodes into all portions of the original copy of the operating

system kernel with the exception of a breakpoint handler in the original copy of the operating system kernel.

24. An apparatus for handling an event in a multiprocessor system, comprising:

a memory; and

a debugger coupled to the memory,

wherein the memory includes an original copy of an operating system kernel and a new copy of the operating system kernel that is in a different physical location of the memory from the physical location of the original copy of the operating system kernel, and

wherein the debugger, in response to an event, switches address mapping of the multiprocessor system to direct mapping to the new copy of the operation system kernel and inserts at least one breakpoint in the original copy of the operating system kernel to generate a modified copy of the operating system kernel.

25. The apparatus of claim 24, wherein the debugger performs one or more actions to cause one or more caches of processors of the multiprocessor system to be flushed.

26. The apparatus of claim 24, wherein the event is encountering one of a breakpoint, a trigger, a watchpoint, or the occurrence of an error.

27. The apparatus of claim 24, wherein the debugger inserts at least one breakpoint by inserting a repeating pattern of breakpoint instruction opcodes into all portions of the original copy of the operating system kernel with the exception of a breakpoint handler in the original copy of the operating system kernel.

28. The apparatus of claim 24, wherein the multiprocessor system is a multiprocessor system that does not support the use of non-maskable interrupts.

29. The apparatus of claim 24, wherein the multiprocessor system is one of a symmetric multiprocessor system and a non-uniform memory access system.